CCS Report 593

A FURTHER GENERALIZED KETTELE ALGORITHM WITH MULTIPLE CONSTRAINTS

by

A. Charnes
D. Zhang

CENTER FOR CYBERNETIC STUDIES
The University of Texas
Austin, Texas 78712

4

CCS Report 593

A FURTHER GENERALIZED KETTELE ALGORITHM WITH MULTIPLE CONSTRAINTS

by

A. Charnes
D. Zhang

March, 1988

The research for this paper was partly supported by National Science Foundation Grant SES 8520806 and Office of Naval Research Contract N00014-86-C-0398, with the Center for Cybernetic Studies at The University of Texas at Austin. Reproduction in whole or in part is permitted for any purpose of the U.S. Government.

CENTER FOR CYBERNETIC STUDIES

A. Charnes, Director

College of Business Administration, 5.202
The University of Texas at Austin
Austin, Texas 78712-1177
(512) 471-1821

A Further Generalized Kettele Algorithm
With Multiple Constraints

A. Charnes
D. Zhang

Abstract

J.D. Kettele [1], using dynamic programming, developed a simple algorithm (KA) for the optimal redundancy problem in reliability and life testing problems with a single constraint. F. Proschan and T.B. Bray [2] gave a generalization of Kettele's dynamic programming algorithm to include multiple constraints. To solve a much broader class of optimization problems than in [1], R.E. Barlow ~~and F.~~ Proschan generalized the Kettele algorithm (GKA) to apply to strictly increasing separable function problems with a single constraint [3].

In this paper, we consider a still more general optimization model and develop a Further Generalized Kettele Algorithm ~~(FGKA)~~ to apply to multiple constraints, etc. As an example, an integer Lexicographic programming model will be solved ~~(corollary 1, section 2)~~. Furthermore, another form of the more general optimization model is pointed out in section 4 of the paper.

Key Words

reliability

life testing

Kettele Algorithms

optimal redundancy

1. Introduction

Kettele [1] presents an algorithm for allocating redundancy so as to maximize system reliability without exceeding a specified cost. Specifically, a system consisting of k "stages" is considered. The system functions if and only if each stage functions. Stage i consists of n_i (to be determined) units of type i in parallel, so that stage functions if and only if at least one of n_i units of type i functions, $i = 1, 2, \ldots, k$. Suppose unit i has a "cost" c_i, $i = 1, \ldots, k$. A unit of type i has probability p_i of functioning, independently of the functioning or nonfunctioning of the other units of the system. Thus system reliability is given by $P(\vec{n}) = \Pi_{i=1}^{i=k}\left(1-(1-p_i)^{n_i}\right)$. Then the problem considered in [1] is

$$\max \prod_{i=1}^{k}\left(1-(1-p_i)^{n_i}\right),$$

$$\text{s.t.} \sum_{i=1}^{k} c_i n_i \le c. \qquad (1)$$

$$n_i \ge 0, \quad \text{integer},$$

where c is a limit "cost".

Proschan and Bray [2] generalize KA to solve the more general problem of maximizing system reliability without exceeding any of several linear constraints, i.e., specifically,

$$\max \prod_{i=1}^{k}\left(1-(1-p_i)^{n_i}\right),$$

$$\text{s.t.} \sum_{i=1}^{k} c_{ij} n_i \le c_j, \quad (j = 1, 2, \ldots, r)$$

$$n_i \ge 0, \quad \text{integer},$$

where c_{ij} is the "cost" of the unit i of the j^{th} type, $i = 1, \ldots, k$; $j = 1, \ldots r$, and c_j is the limit "cost" of the j^{th} type. As an example, the first type of cost might be money, the second weight, the third volume, the fourth population rate.

Barlow and Proschan [3] then give a Generalized Kettele Algorithm to solve an optimization model more general than in [1] as follows. Suppose $x_1, \ldots, x_k$ are k variables called "decision variables"; $x_i \in S_i = \left\{x_i^{(1)} < x_i^{(2)} < \ldots\right\}$, $i = 1, \ldots, k$. Let $\vec{x}_i = (x_1, \ldots, x_i)$, $i = 1, \ldots, k$. Assume $f_i, \ldots, f_k$ are strictly increasing functions, with $y_1(x_1) = f_1(x_1)$, $y_2(\vec{x}_2) = f_2(y_1, x_2)$, $y_3(\vec{x}_3) = f_3(y_2, x_3), \ldots, y_k(\vec{x}_k) = f_k(y_{k-1}, x_k)$. Similarly, assume $g_1, \ldots, g_k$ are strictly increasing functions, with $z_1(x_1) = g_1(x_1)$, $z_2(\vec{x}_2) = g_2(z_1, x_2)$, $z_3(\vec{x}_3) = g_3(z_2, x_3), \ldots, z_k(\vec{x}_k) = g_k(z_{k-1}, x_k)$. It is suggestive to call $\vec{x}_i$ an "allocation" of order i, y_i a "payoff" of order i, and z_i a "cost" of order i ($i = 1, \ldots, k$) even though the model is more general than these terms would indicate.

The Barlow and Proschan (B-P) model is

$$\begin{aligned} &\max \quad y_k(\vec{x}_k), \\ &\text{s.t.} \quad z_k(\vec{x}_k) \le c, \\ &\vec{x}_k = (x_1, \ldots, x_k) \in S_1 x, \ldots, x S_k, \end{aligned} \tag{2}$$

where c is a limit "cost". In this model there is only a single constraint. The cost z_k may not be a sum of linear functions $c_j x_j$, $j = 1, \ldots, k$. It is strictly increasing in the number of spares of each type.

B-P gave some examples (see examples and exercises, [3]) and pointed out that a great many other models requiring optimization subject to a constraint arising in reliability, and more generally in operations research, are special cases of the general optimization model. Such problems may be solved by the Generalized Kettele Algorithm [3].

In this paper, we propose a more general optimization model and develop an associated Further Generalized Kettele Algorithm (FGKA). Thereby, the optimal problems which can be solved by FGKA will not confine to a simple numerical objective function or a single constraint.

<u>Example 1:</u> An Integer Lexicographic Problem

$$\max \quad \left(y_k^{(1)}(\vec{x}_k), y_k^{(2)}(\vec{x}_k), \ldots, y_k^{(l)}(\vec{x}_k)\right),$$

s.t.

$$z_k^{(i)}(\vec{x}_k) \le c_i, \quad i = 1, 2, \ldots, m, \tag{3}$$

where c_i is a constant, $i = 1, \ldots, m$; $\vec{x}_i = (x_1, \ldots, x_i)$, and x_j is a real variable, $i, j = 1, \ldots, k$; $f_1^{(j)}, \ldots, f_k^{(j)}$ all are strictly increasing functions with real values, and $y_1^{(j)}(x_1) = f^{(j)}(x_1)$, $y_2^{(j)}(\vec{x}_2) = f_2^{(j)}\left(y_1^{(j)}, x_2\right), \ldots, y_k^{(j)}(\vec{x}_k) = f_k^{(j)}\left(y_{k-1}^{(j)}, x_k\right)$, $j = 1, \ldots, l$; $g_1^{(j)}, \ldots, g_k^{(j)}$ all are strictly increasing functions with real values, and $z_1^{(j)}(x_1) = g_1^{(j)}(x_1)$, $z_2^{(j)}(\vec{x}_2) = g_2^{(j)}\left(z_1^{(j)}, x_2\right), \ldots,$ $z_k^{(j)}(\vec{x}_k) = g_k^{(j)}\left(z_{k-1}^{(j)}, x_k\right)$, $z_{i-1}^{(j)} = g_i^{(j)}(z_{i-1}^{(j)}, x_i^{(1)})$, $i \geq 2$, $j = 1, \ldots, m$; $\left(y_k^{(1)}, \ldots, y_k^{(l)}\right)$ belongs to l-dimensional real vector space R^l with the Lexicographic ordering.

The Lexicographic problem is a special case of our still more general optimiztion model. And as we know, an integer bottleneck problem and an integer time-cost problem are in some sense special cases of the integer Lexicographic problems.

2. Still More General Optimization Model and FGKA

Now, let us give the still more general optimization model. Suppose S_i is a countable well ordered set, $S_i = \left\{x_i^{(1)} < x_i^{(2)} < \ldots\right\}$, $i = 1, 2, \ldots, k$. Y, Z both are ordered sets. In Z there exists a maximal element z_0 (or ∞). Let x_i represent a variable called a "decision" variable, $x_i \in S_i$, and $\vec{x}_i = (x_1, x_2, \ldots, x_i)$. Let $f_1, f_2, \ldots, f_k; g_1, g_2, \ldots, g_k$ all be strictly increasing functions, with $y_1(x_1) = f_1(x_1)$, $y_2(\vec{x}_2) = f_2(y_1, x_2), \ldots, y_k(\vec{x}_k) = f_k(y_{k-1}, x_k)$ and $y_i \in Y$; $z_1(x_1) = g_1(x_1)$, $z_2(\vec{x}_2) = g_2(z_1, x_2), \ldots, z_k(\vec{x}_k) = g_k(z_{k-1}, x_k)$ and $z_i \in Z$, $i = 1, 2, \ldots, k$; for $i > 1$, $g_i(z_0, x_i) = z_0$, $z_{i-1} = g_i(z_{i-1}, x_i^{(1)})$.

Call $\vec{x}_i$ an allocation of order i, y_i a payoff of order i, and z_i a cost of order i, $i = 1, 2, \ldots k$.

The still more general optimization model is

$$
\begin{aligned}
&\text{Max } y_k(\vec{x}_k), \\
&\text{s.t.} \\
&\qquad z_k(\vec{x}_k) \le c,
\end{aligned}
\tag{4}
$$

where $c \in Z \backslash \{z_0\}$, $\vec{x}_k = (x_1, x_2, \ldots, x_k)$, $x_j \in S_j$, $j = 1, 2, \ldots, k$.

Assume that (4) and $z_j > c$, $j = 1, \ldots, k$ all have solutions.

<u>Definition1:</u> Allocation $\vec{x}_i$ dominates allocation $\vec{x}_i'$ if

(i) $y_i(\vec{x}_i) > y_i(\vec{x}_i')$ and $z_i(\vec{x}_i) \le z_i(\vec{x}_i') < z_0$, or

(ii) $y_i(\vec{x}_i) = y_i(\vec{x}_i')$ and $z_i(\vec{x}_i) < z_i(\vec{x}_i') < z_0$, or

(iii) $z_i(\vec{x}_i') = z_0$

We write $\vec{x}_i \overset{\Delta}{>} \vec{x}_i'$. We also say the corresponding payoff-cost pair (y_i, z_i) dominates (y_i', z_i') and write $(y_i, z_i) \overset{\Delta}{>} (y_i', z_i')$.

<u>Definition 2:</u> $\vec{x}_i$ is called an undominated allocation of order i if there exists no $\vec{x}_i'$ such that $\vec{x}_i' \overset{\Delta}{>} \vec{x}_i$. We also say the corresponding payoff-cost pair (y_i, z_i) is undominated.

Definition 3: A complete sequence of undominated allocations of order i (ending in $\vec{x}_i^{(S)}$, say) is a sequence of undominated allocations $\vec{x}_i^{(1)}, \ldots, \vec{x}_i^{(s)}$ each of order i, such that

(i) $y_i\left(\vec{x}_i^{(1)}\right) \leq \ldots \leq y_i\left(\vec{x}_i^{(s)}\right)$,

(ii) $z_i\left(\vec{x}_i^{(1)}\right) \leq \ldots \leq z_i\left(\vec{x}_i^{(s)}\right)$, and

(iii) If $\vec{x}_i$ is undominated and yields a payoff-cost pair distinct from those of $\vec{x}_i^{(1)}, \ldots, \vec{x}_i^{(s)}$, then $y_i(\vec{x}_i) \geq y_i\left(\vec{x}_i^{(s)}\right)$ and $z_i(\vec{x}_i) \geq z_i\left(\vec{x}_i^{(s)}\right)$.

We call the corresponding sequence $\left(y_i^{(1)}, z_i^{(1)}\right), \ldots, \left(y_i^{(s)}, z_i^{(s)}\right)$ a complete undominated sequence of payoff-cost pairs of order i.

We now present an algorithm to solve this more general optimization model (4). We shall call it the Further Generalized Kettele Algorithm (FGKA).

Step 1: Compute $y_1^{(i)} = f_1\left(x_1^{(i)}\right), z_1^{(i)} = g_1\left(x_1^{(i)}\right)$ for $i = 1, 2, \ldots, s_1$, where $z_1^{(s_1)} \leq c$, and $z_1^{(s_1+1)} > c$. There exists such s_1 since $z_1(x_1) = z_k(x_1, x_2^{(1)}, \ldots, x_k^{(1)})$ and $z_k \leq c$, $z_1 > c$ both have solutions. $\left(y_1^{(1)}, z_1^{(1)}\right), \ldots, \left(y_1^{(s_1)}, z_1^{(s_1)}\right)$ constitute a complete sequence of undominated payoff-cost pairs of order 1 not violating the cost constraint.

Step 2: Compute $y_{2ij} = f_2\left(y_1^{(i)}, x_2^{(j)}\right), z_{2ij} = g_2\left(z_1^{(i)}, x_2^{(j)}\right)$ for $i = 1, 2, \ldots, s_1$; $j = 1, 2, \ldots$ such that $z_{2ij} \leq c$. Enter the payoff-cost pair (y_{2ij}, z_{2ij}) in row i, column j of a table of payoff-cost pairs.

Step 3: The lowest cost undominated pair $\left(y_2^{(1)}, z_2^{(1)}\right)$ in the table is clearly (y_{211}, z_{211}) if $z_{211} \leq c$. If $z_{211} > c$, there is no solution.

To determine $\left(y_2^{(2)}, z_2^{(2)}\right)$, find (y_{2ij}, z_{2ij}) such that $y_{2ij} > y_2^{(1)}$ with z_{2ij} minimum among entries satisfying this inequality. If several entries of identical cost qualify, choose the highest payoff y_{2ij}. If several entries of identical cost and identical highest payoff qualify, choose one at random, since all are equally cost-effective. The payoff-cost entry so chosen is $\left(y_2^{(2)}, z_2^{(2)}\right)$.

Continue in this fashion, i.e. having found $\left(y_2^{(r)}, z_2^{(r)}\right)$, then $\left(y_2^{(r+1)}, z_2^{(r+1)}\right)$ is the pair (y_{2ij}, z_{2ij}) such that $y_{2ij} > y_2^{(r)}$ with z_{2ij} minimum among pairs satisfying this inequality. In case of ties, follow the rules for breaking ties described above.

Stop at $\left(y_2^{(s_2)}, z_2^{(s_2)}\right)$, where s_2 is determined so that $z_2^{(s_2)} \le c$, while $z_2^{(s_2+1)} > c$. Such s_2 exists if there is a solution to (4).

The sequence $\left(y_2^{(1)}, z_2^{(1)}\right), \ldots, \left(y_2^{(s_2)}, z_2^{(s_2)}\right)$ obtained this way constitutes a complete sequence of undominated payoff-cost pairs of order 2 not exceeding the cost limit c.

Step 4: Proceeding in a similar fashion using the payoff-cost pairs (y_{3ij}, z_{3ij}), where $y_{3ij} = f_3\left(y_2^{(i)}, z_3^{(j)}\right)$, $z_{3ij} = g_3\left(z_2^{(i)}, x_3^{(j)}\right)$ for $i = 1, 2, \ldots, s_2$; $j = 1, 2, \ldots$, such that $z_{3ij} \le c$, obtain a complete sequence of payoff-cost pairs not violating the cost constraint. Continue in this fashion until at the k^{th} stage, arrive at the complete sequence of undominated payoff-cost pairs $\left(y_k^{(1)}, z_k^{(1)}\right), \ldots, \left(y_k^{(s_k)}, z_k^{(s_k)}\right)$, where $z_k^{(s_k)} \le c$, while $z_k^{(s_k+1)} > c$.

Step 5: Finally, $y_k^{(s_k)}$ is the maximum payoff achievalbe under the cost constraint; the corresponding cost is $z_k^{(s_k)}$. The allocation $\vec{x}^{(s_k)}$ yielding the payoff-cost pair $\left(y_k^{(s_k)}, z_k^{(s_k)}\right)$ is the solution to (4).

Theorem 1. The payoff-cost pair $\left(y_k^{(s_k)}, z_k^{(s_k)}\right)$ and the allocation $\vec{x}^{(s_k)}$ obtained using the FGKA is the solution to the more general optimal model (4). That is, it is the maximum payoff achievable with corresponding cost $z_k^{(s_k)} \le c$, the cost constraint.

Proof. The argument of the proof is similar to that of the original theorem about the GKA (p. 221, [3]). In the original proof two lemmas are used. They are as follows.

Lemma 4.5. (a) Every payoff-cost pair of order 2 obtained by GKA is undominated. (b) Every undominated payoff-cost pair of order 2 may be obtained by the GKA.

Lemma 4.6. Let $(y_2, z_2) \overset{\Delta}{>} (y_2', z_2')$, $y_3 = f_3(y_2, x_3)$, $z_3 = g_3(z_2, x_3)$, $y_3' = f_3(y_2', x_3)$, and $z_3' = g_3(z_2', x_3)$. Then $(y_3, z_3) \overset{\Delta}{>} (y_3', z_3')$.

The two lemmas are correct for FGKA. The proof of the former is almost word for word repeat of the original lemma proof. The proof of the latter is little different from the original. By the hypothesis of Lemma 4.6,

(i) $y_2 > y_2'$ and $z_2 \leq z_2' < z_o$, or

(ii) $y_2 = y_2'$ and $z_2 < z_2' < z_o$, or

(iii) $z_2' = z_o$.

In cases (i) and (ii), we know $(y_3, z_3) \overset{\Delta}{>} (y_3', z_3')$. In case (iii), this also holds since g_3 is monotone and $z_3' = g_3(z_2', x_3) = z_o$. Hence, Lemma 4.6 holds for FGKA.

Using the two new lemmas and repeating the argument of the proof of the original theorem 4.7 word by word, we get a proof of Theorem 2.

Q.E.D.

Now we show that the general optimization model (B-P) under multiple constraints is a special case of the still more general optimization model (4).

Theorem 2. The problem

$$\max \quad y_k(\vec{x}_k),$$

s.t.

$$z_k^{(i)}(\vec{x}_k) \leq c_i, \quad i = 1, 2, \ldots, m, \qquad (5)$$

$$\vec{x}_k = (x_1, x_2, \ldots, x_k) \in S_1 \times S_2 \times \ldots \times S_k,$$

where S_j, y_i, x_j, $\vec{x}_j$, Y_j, $z_j^{(i)}$ all are as in the more general optimization model; $Z^{(i)}$ is an ordered set without assuming a maximal element, is a special case of the still more general optimization model (4).

Proof. Let $Z_o = Z^{(1)} \times Z^{(2)} \times \ldots \times Z^{(m)}$. Define an equivalence relation $\equiv$ among elements of Z_o as follows. For $a_i, b_i \in Z^{(i)}$, $i = 1, 2, \ldots, m$,

$$(a_1, a_2, \ldots, a_m) \equiv (b_1, b_2, \ldots, b_m)$$

if and only if either

(i) $a_i > c_i$ for some $i = 2, 3, \ldots, m$, and $b_j > c_j$ for some $j = 2, 3, \ldots, m$, or

(ii) $a_i \le c_i$ and $b_i \le c_i$, $i = 2, 3, \ldots, m$, and $a_1 = b_1$.

Thus we get a collection Z of classes of equivalent elements of Z_0. Denote the class including $(c_1, c_2, \ldots, c_m)$ by c, $\left(z_k^{(1)}(\vec{x}_k), z_k^{(2)}(\vec{x}_k), \ldots, z_k^{(m)}(\vec{x}_k)\right)$ by $z_k(\vec{x}_k)$. Define

$(a_1, a_2, \ldots, a_m) > (b_1, b_2, \ldots, b_m)$

if and only if either

(i) $(a_1, a_2, \ldots, a_m) = z_0 \ne (b_1, b_2, \ldots, b_m)$, where z_0 is the class whose one of components $2, 3, \ldots, m$ is greater than corresponding c_i, or

(ii) $z_0 \ne (a_1, a_2, \ldots, a_m)$ or $(b_1, b_2, \ldots, b_m)$, where $a_1 > b_1$.

Then Z is an ordered set with a maximal element z_0.

Here we do not distinguish a clas^ from its representatives.

So, (5) is equivalent to a (4) in which $z_k \in Z$, $c \in Z \backslash \{z_0\}$. That is, (5) is a special case of (4).

Q.E.D.

Similarly, we can prove corollary 1. In fact, it can be obtained as a conclusion of Theorem 1 itself, also.

<u>Corollary 1</u>. The integer Lexicographic problem (3) is a special case of the more general optimization model (4), or (5).

3. An Example using FGKA

Example 2. Fill Rate Model (compare it to [3], p. 209)

$$\max \quad R(\vec{n}),$$

s.t.

$$\sum_{i=1}^{4} n_i c_i \le c_o,$$

$$\sum_{i=1}^{4} n_i w_i \le w_o, \qquad (6)$$

$$n_i \ge 0, \text{ integer}, \quad i = 1, 2, 3, 4,$$

where $\vec{n} = (n_1, n_2, n_3, n_4)$ is a spares allocation; c_i is the price of part i, c_o is a spares budget constraint; w_i is the weight of part i, w_o is the limit weight of a spare parts kit, $R(\vec{n})$ is the fill rate,

$$R(\vec{n}) = \sum_{i=1}^{4} \lambda_i \sum_{j=0}^{n_i-1} \frac{(\lambda_i \upsilon_i)^j}{j!} e^{-\lambda_i \upsilon_i} \left(\sum_{i=1}^{4} \lambda_i \right)^{-1},$$

where $\sum_{j=0}^{-1} \equiv 0$. Here we assume that

(i) Demands for spares of type i at a maintenance depot is governed by a Poisson process with demand rate λ_i.

(ii) Enough repair facilities are available at the depot so that repair of a failed unit is initiated as soon as it is received.

(iii) The mean time to repair a failed unit of type i is υ_i.

(iv) The cost of purchasing n_i units of type i is $n_i c_i$.

(v) There are k part types, i.e., $i = 1, 2, \ldots, k$.

In (6) there are two constraints, which says, that the budget is not to be exceeded nor is the allowed total weight (or volume).

To solve the example, first let $S_i = \{0, 1, 2, \ldots\}$, Y the set R^1 of real numbers. To define Z, first define an equivalence relation $\equiv$ in the 2-dimensional real vector space R^2 as follows.

$$(u_1, u_2) \equiv (v_1, v_2)$$

if and only if either

(i) $u_1 > w_o$ and $v_1 > w_o$, or

(ii) $u_1, v_1 \leq w_o$ and $u_2 = v_2$

Denote (u_1, u_2) by z_o if $u_1 > w_o$. Then let Z be the collection of equivalence classes of elements in R^2. Define an order relation $>$ in Z:

$$(u_1, u_2) > (v_1, v_2)$$

if and only if either

(i) $u_1 > w_o$ and $v_1 \leq w_o$, or

(ii) $u_1, v_1 \leq w_o$ and $u_2 > v_2$.

Denote (w_o, c_o) by c.

The decision variables are to be the number of spares of the k different types, (n_1, n_2, n_3, n_4). The functions of f_i and g_i of the more general model are given by

$$f_1(n_1) = \lambda_1 \sum_{j=0}^{n_1-1} \frac{(\lambda_1 \upsilon_1)^j}{j!} e^{-\lambda_1 \upsilon_1}$$

$$f_i(R, n_i) = R + \lambda_i \sum_{j=0}^{n_i-1} \frac{(\lambda_i \upsilon_i)^j}{j!} e^{-\lambda_i \upsilon_i}$$

$$g_1(n_1) = (w_1 n_1, c_1 n_1),$$

$$g_i((w_1\ c)), n_i) = (w, c) + (w_i\ n_i, c_i\ n_i).$$

So, all f_i are strictly increasing functions. And all g_i are increasing in Z, strictly increasing in $Z \backslash (z_o\}$.

For example, consider the situation of a budget $c_o = 1500$ (in dollars), weight limit $w_o = 100$ (in kg), and demand, repair, weight and cost data as shown in Table 1.

Part Type, i	Demand Rate (per hour) λ_i	Mean Time to Repair Failed Part υ_i	kg weight of part, w_i	Dollar Cost of part, c_i	computed value: $\lambda_i \upsilon_i$
1	.01	100	30	200	1.0
2	.02	150	10	100	3.0
3	.03	60	40	300	1.8
4	.01	200	5	250	2.0

Table 1. Input Data for Spares Allocation

To simplify the computation throughout, we drop the denominator to $\sum_1^k \lambda_i$ in $R(\vec{n})$ since it is constant throughout and just compute the fill rate numerator $\sum_{i=1}^k \lambda_i R(n_i)$, where

$$R_i(n) = \sum_{j=0}^{n-1} \frac{(\lambda_i \upsilon_i)^j}{j!} e^{-\lambda_i \upsilon_i} \qquad i = 1, 2, \ldots, k$$

The steps taken in carrying out FGKA are the following.

(1) Noting that $n_i = 0, 1, 2, \ldots$, in Step 1 we get

$$f_1(n_1) = \lambda_1 \sum_{j=0}^{n_1-1} \frac{(\lambda_1 \upsilon_1)^j}{j!} e^{-\lambda_1 \upsilon_1} = .01 \sum_{j=0}^{n_1-1} \frac{(1.0)^j}{j!} e^{-1.0}$$

and

$$g_1(n_1) = (w_1 n_1, c_1 n_1) = (30n_1, 200n_1),$$

and

$$c = (w_o, c_o) = (100, 1500)$$

we get the complete sequence of undominated payoff-cost pairs of order 1,

$(y_1{}^{(1)},\ z_1{}^{(1)}) = (0,\ (0,\ 0)),$ $\qquad (y_1{}^{(2)},\ z_1{}^{(2)}) = (.00368,\ (30,\ 200)),$

$(y_1{}^{(3)},\ z_1{}^{(3)}) = (.00736,\ (60,\ 400)),$ $\qquad (y_1{}^{(4)},\ z_1{}^{(4)}) = (.00920,\ (90,\ 600)).$

Here $s_1 = 4$

(2) Noting that $n_2 = 0, 1, 2, \ldots,$ in Step 2 and Step 3,

$$f_2(R_1,\ n_2) = R + \lambda_2 \sum_{j=0}^{n_2-1} \frac{(\lambda_2 \upsilon_2)^j}{j!} e^{-\lambda_j \upsilon_j} = R + .02 \sum_{j=0}^{n_2-1} \frac{(3.0)^j}{j!} e^{-3.0}$$

and

$$g_2((w,c),\ n_2) = (w,\ c) + (10\ n_2\ ,\ 100\ n_2),$$

we get Table 2.

The complete sequence of undominated payoff-cost pairs of order 2 is shown by •——•——•— ··· ——• in Table 2. Here $s_2 = 9$.

(3) Since $n_3 = 0, 1, 2, \ldots,$ in Step 4

$$f_3(R,\ n_3) = R + \lambda_j \sum_{j=0}^{n_3-1} \frac{(\lambda_3 \upsilon_3)^j}{j!} e^{-\lambda_j \upsilon_j} = R + .03 \sum_{j=0}^{n_3-1} \frac{(1.8)^j}{j!} e^{-1.8}$$

and

$$g_3((w,\ c),\ n_3) = (w,\ c) + (40\ n_3\ ,\ 300\ n_3),$$

we obtain Table 3. Here $s^{(3)} = 9$, i.e., the complete sequence of undominated payoff-cost pairs of order 3 shown in Table 3 has 9 elements.

(4) Noting that $n_4 = 0, 1, 2, \ldots,$ in Step 4

$$f_4(R,\ n_4) = R + \lambda_4 \sum_{j=0}^{n_4-1} \frac{(\lambda_4 \upsilon_4)^j}{j!} e^{-\lambda_4 \upsilon_4} = R + .01 \sum_{j=0}^{n_4-1} \frac{(2.0)^j}{j!} e^{-2.0}$$

and

$$g_4((w,\ c),\ n_4) = (w,\ c) + (5\ n_4\ ,\ 250\ n_4),$$

Thus we get Table 4.

The complete sequence of undominated payoff-cost pairs of order 4 is

0 , 0, 0, 0; 0, 1, 0, 0; 0, 2, 0, 0; 0, 3, 0, 0;

0, 4, 0, 0; 0, 5, 0, 0; 1, 4, 0, 0; 1, 5, 0, 0;

2, 4, 0, 0; 2, 4, 0, 1; 2, 4, 0, 2; 1, 5, 0, 3.

(5) Finally, a solution of example 3 is

$(n_1, n_2, n_3, n_4) = (1, 5, 0, 3).$

with Max $(\vec{n})$ = .02576.

$n_2^{(j)}$

$z_1^{(i)}$

NS FRN WC	0 0 (0, 0)	1 .00010 (10,100)	2 .00548 (20,200)	3 .00996 (30,300)	4 .01332 (40,400)	5 .01533 (50,500)	6 .01634 (60,600)	7 .01677 (70,700)	8 .01694 (80,800)	9 .01699 (90,900)	10 .01701 (100,1000)
0 0 (0, 0)	0, 0 0 (0, 0)	0, 1 .00010 (10,100)	0, 2 .00548 (20,200)	0, 3 .00996 (30,300)	0, 4 .01332 (40,400)	0, 5 .01533 (50,500)	0, 6 .01634 (60,600)	0, 7 .01677 (70,700)	0, 8 .01694 (80,800)	0, 9 .01699 (90,900)	0, 10 .01701 (100,1000)
1 .00368 (30,200)	1, 0 .00368 (30,200)	1 ,1 .00468 (40,300)	1, 2 .00916 (50,400)	1, 3 .01304 (60,500)	1, 4 .01700 (70,600)	1, 5 .01901 (80,700)	1, 6 .02002 (90,800)	1, 7 .02045 (100,900)			
2 .00736 (60,400)	2, 0 .00736 (60,400)	2, 1 .00834 (70,500)	2, 2 .01284 (80,600)	2, 3 .01732 (90,700)	2, 4 .02068 (100,800)						
3 .00920 (90,600)	3, 0 .00920 (90,600)	3,1 .01020 (100,700)									

Table 2 (y_{2ij}, z_{2ij})

Notation: NS - number of spares, FRN - fill rate numerator, WC - weight and cost.

$n_3^{(i)}$

$z_2^{(i)}$

NS FRN WC	0 0 (0, 0)	1 .004959 (40, 300)	2 .013885 (80,600)
0, 0 0 (0, 0)	0, 0, 0 0 (0, 0)	0, 0, 1 .00496 (40, 300)	0, 0, 2 .01389 (80,600)
0, 1 .00100 (10,100)	0, 1, 0 .00100 (10,100)	0, 1, 1 .00596 (50,400)	0, 1, 2 .01489 (90,700)
0, 2 .00548 (20,200)	0, 2, 0 .00548 (20,200)	0, 2, 1 .01044 (60,500)	0, 2, 2 .01937 (100,800)
0, 3 .00996 (30,300)	0, 3, 0 .00996 (30,300)	0, 3, 1 .01492 (70,600)	
0, 4 .01332 (40,400)	0, 4, 0 .01332 (40,400)	0, 4, 1 .01828 (80,700)	
0, 5 .01533 (50,500)	0, 5, 0 .01533 (50,500)	0, 5, 1 .02029 (90,800)	
1, 4 .01700 (70,600)	1, 4, 0 .01700 (70,600)		
1, 5 .01901 (80,700)	1, 5, 0 .01901 (80,700)		
2, 4 .02068 (100,800)	2, 4, 0 .02068 (100,800)		

Table 3 (y_{3ij}, z_{3ij})

$n_4^{(j)}$ (columns); $z_3^{(i)}$ (rows)

NS FRN WC	0 0 (0, 0)	1 .00135 (5, 250)	2 .00405 (10,500)	3 .00675 (15,750)	4 .00855 (20,1000)	5 .00945 (25,1250)	6 .00981 (30,1500)
0, 0, 0 0 (0, 0)	0, 0,0,0 0 (0, 0)	0, 0,0, 1 .00135 (5, 250)	0, 0,0,2 .00405 (10,500)	0,0,0,3 .00675 (15,750)	0, 0,0,4 .00855 (20,1000)	0, 0,0,5 .00945 (25,1250)	0,0,0,6 .00981 (30,1500)
0,1,0 .00100 (10,100)	0,1,0,0 .00100 (10,100)	0,1,0,1 .00235 (15,350)	0,1,0,2 .00505 (20,600)	0,1,0,3 .00775 (25,850)	0,1,0,4 .00955 (30,1100)	0,1,0,5 .01045 (35,1350)	
0,2,0 .00548 (20,200)	0,2,0,0 .00548 (20,200)	0,2,0,1 .00683 (25,450)	0,2,0,2 .00953 (30,700)	0,2,0,3 .01223 (35,950)	0,2,0,4 .01403 (40,1200)	0,2,0,5 .01493 (45,1450)	
0,3,0 .00996 (30,300)	0,3,0,0 .00996 (30,300)	0,3,0,1 .01131 (35,550)	0,3,0,2 .01401 (40,800)	0,3,0,3 .01671 (45,1050)	0,3,0,4 .01851 (50,1300)		
0,4,0 .01332 (40,400)	0,4,0,0 .01332 (40,400)	0,4,0,1 .01467 (45,650)	0,4,0,2 .01737 (50,900)	0,4,0,3 .02007 (55,1150)	0,4,0,4 .02187 (60,1400)		
0,5,0 .01533 (50,500)	0,5,0,0 .01533 (50,500)	0,5,0,1 .01668 (55,750)	0,5,0,2 .01938 (60,1000)	0,5,0,3 .02208 (65,1250)	0,5,0,4 .02388 (70,1500)		
1,4,0 .01700 (70,600)	1,4,0,0 .01700 (70,600)	1,4,0,1 .01835 (65,850)	1,4,0,2 .02105 (70,1100)	1,4,0,3 .02375 (75,1350)			
1,5,0 .01901 (80,700)	1,5,0,0 .01901 (80,700)	1,5,0,1 .02036 (75,950)	1,5,0,2 .02306 (80,1200)	1,5,0,3 .02576 (85,1450)			
2,4,0 .02068 (100,800)	2,4,0,0 .02068 (100,800)	2,4,0,1 .02203 (85,1050)	2,4,0,2 .02473 (90,1300)				

Table 4 $(y_{4ij},\ z_{4ij})$

4. Another Form of the More General Optimization Model

Corresponding to (1), consider an optimal design problem

$$\min \sum_{i=1}^{k} c_i n_i$$

$$\text{s.t.} \quad \prod_{i=1}^{k}\left(1-(1-p_i)^{n_i}\right) \geq R, \tag{7}$$

$$n_i \geq 0, \text{ integer},$$

where R is a limit reliability, c_i, n_i, p_i are as in (1).

Generally, let us consider a resolvable problem

$$\min \quad y_k(\vec{x}_k),$$

$$\text{s.t.} \quad z_k(\vec{x}_k) \geq c, \tag{8}$$

where $\vec{x}_k$, y_k, z_k and c are as in (4). The only difference is that in Z there exists a minimal element z_0, but a maximal element $z_j < c$ has a solution, $j = 1, \ldots, k$. And

$$S_j = \{\ldots < x_j^{(2)} < x_j^{(1)}\}, \quad 1, 2, \ldots, k. \tag{9}$$

For (7), select positive integers $m_1, \ldots, m_k$ large enough. Let $s_j = \{0 < \ldots < m_j - 1 < m_j\}$, $y_1(x_1) = f_1(x_1) = c_1x_1$, $y_2(\vec{x}_2) = f_2(y_1, x_2) = y_1 + c_2x_2, \ldots, y_k(\vec{x}_k) = f_k(y_{k-1}, x_k) = y_{k-1} + c_k x_k$; $z_1(x_1) = (1-(1-p_1)^{x_1})$, $z_2 = z_1(1-(1-p_2)^{x_2})\dfrac{1}{(1-(1-p_2)^{m_2}}, \ldots,$ $z_k = z_{k-1}(1-(1-p_k)^{x_k})\dfrac{1}{(1-(1-p_k)^{m_k}}$. Then (7) is an example of (8).

We point out that (8) is another form of the more general optimization model (4). Let $S_i^* = \{x_i^{(1)} \lhd x_i^{(2)} \lhd \ldots\}$ be an ordered set with ordering $\lhd$, $i = 1, \ldots, k$. That is $x_i^{(u)} \lhd x_i^{(v)}$ if and only if $x_i^{(u)} > x_i^{(v)}$. Define a new ordering $\lhd$ in Y such that $a \lhd b$ if and only if $a > b$. Denote the Y with the new relation $\lhd$ by Y^*. Similarly, we get Z^*. Thus z_0 is a maximum in Z^* about the relation $\lhd$. Now, the problem (8) gets a new form as follows.

$$\max \quad y_k(\vec{x}_k),$$

$$\text{s.t.} \quad z_k(\vec{x}_k) \trianglelefteq c, \tag{10}$$

where the sign max is about the new relation $\lhd$ in Y^* and the equality sign is the same with the original.

Theorem 3. The problem (8) is just another form of the still more general optimization model (4).

Hence, we can use the FGKA solving (8) or (10).

Furthermore, let us look at another optimal design problem

$$\begin{aligned} \min \quad & \sum_{i=1}^{k} c_i n_i \\ \text{s.t.} \quad & \prod_{i=1}^{k}\left(1-(1-p_i)^{n_i}\right) \geq R \\ & \sum_{i=1}^{k} w_i n_i \leq w, \end{aligned} \tag{11}$$

where w_i is the weight of a part of type i, w is the total weight limit. Other signs are as in (7).

Generally, we would consider a problem

$$\begin{aligned} \min \quad & y_k(\vec{x}_k), \\ \text{s.t.} \quad & z_k(\vec{x}_k) \geq c \\ & z_k'(\vec{x}_k) \leq c', \end{aligned} \tag{12}$$

where $\vec{x}_k$, y_k, z_k, c are as in (8), $z_k' \in Z'$, an ordered set with a maximal element z_0', and $c' \in Z' - \{z_0'\}$.

Theorem 4. The problem

$$\begin{aligned} \min \quad & y_k(\vec{x}_k), \\ \text{s.t.} \quad & z_k^{(i)}(\vec{x}_k) \geq c_i, & i = 1, \ldots, m, \\ & z'^{(i)}_k(\vec{x}_k) \leq c'_i & i = 1, \ldots, n, \\ & \vec{x}_k = (x_1, \ldots, x_k) \in S_1 \times \ldots \times S_k, \end{aligned} \tag{13}$$

where S_j, Y, x_j, $\vec{x}_j$, y_j, all are as in theorem 2, $z'^{(i)}_j$ is as $z_j^{(i)}$ in theorem 2. The corresponding g_j' takes values in the corresponding set Z'; $z_j^{(i)}(\vec{x}_j) = g_j^{(i)}(z_{j-1}^{(i)}, x_j)$ is increasing and takes values in an ordered set Z with minimum z_0, is a special case of (12).

Proof. Let $Z_o = Z^{(1)} x \ldots x Z^{(m)}$. Define an equivalence relation $\equiv$ among elements of Z_o as follows. For $a_i, b_i, \in Z^{(i)}$, $i = 1, \ldots, m$, $(a_1, \ldots, a_m) \equiv (b_1, \ldots b_m)$ if and only if either

(i) $a_i < c_i$ for some $i = 2, \ldots, m$ and $b_j < c_j$ for some $j = 2, \ldots, m$, or

(ii) $a_i \geq c_i$ and $b_i \geq c_i$, $i = 2, \ldots, m$, and $a_1 = b_1$.

Thus we get a collection Z of classes of equivalent elements of Z_o. Denote the class including $(c_1, \ldots, c_m)$ by c, $(z_k^{(1)}(\vec{x}_k), \ldots z_k^{(m)}(\vec{x}_k))$ by $z_k(\vec{x}_k)$. Define $(a_1, \ldots a_m) < (b_1, \ldots, b_m)$ if and only if either

(i) $(a_1, \ldots, a_m) = z_o \neq (b_1, \ldots, b_m)$, where z_o is the class whose one of components $2, 3, \ldots, m$ is less than corresponding c_i, or

(ii) $z_o \neq (a_1, \ldots, a_m)$ or $(b_1, \ldots, b_m)$, where $a_1 < b_1$.

Then Z is an ordered set with a minimum z_o. Similarly, let $Z_o' = Z'^{(1)} x \ldots x Z'^{(n)}$. Define an equivalence relation $\equiv$ among elements of Z_o' as follows. For $a_i, b_i \in Z'^{(i)}$, $i = 1, \ldots, n$, $(a_1, \ldots a_n) \equiv (b_1, \ldots b_n)$ if and only if either

(i) $a_i > c_i'$ for some $i = 2, \ldots, n$ and $b_j > c_j'$ for some $j = 2, \ldots, n$, or

(ii) $a_i \leq c_i'$, $b_i \leq c_i'$, $i = 2, \ldots, n$, and $a_1 = b_1$.

Hence we get a collection Z' of classes of equivalent elements of Z_o'. Define the class including $(c_1', \ldots, c_n')$ by c' and $(z_k'^{(1)}(\vec{x}_k), \ldots z_k'^{(n)}(\vec{x}_k))$ by $z_k'(\vec{x}_k)$.

Define $(a_1, \ldots, a_m) > (b_1, \ldots, b_n)$ if and only if either

(i) $(a_1, \ldots, a_n) = z_o' \neq (b_1, \ldots, b_n)$, where z_o' is the class whose one of components $2, \ldots, n$ is greather than corresponding c_i', or

(ii) $z_o' \neq (a_1, \ldots, a_n)$ or $(b_1, \ldots, b_n)$, where $a_1 > b_1$.

Then Z' is an ordered set with a maximal element z_o'.

Therefore (13) is equivalent to a (12) in which $z_k \in Z$, $c \in Z\backslash\{z_o\}$, $z_k' \in Z'$, $c' \in Z'\backslash\{z_o'\}$. That is, (13) is a special case of (12).

Q.E.D.

REFERENCES

[1] Kettele, J.D., "Least-cost Allocation of Reliability Investment," Operations Research 10, 249-265, 1962.

[2] Proschan, F. and Bray, T.A., "Optimal Redundancy Under Multiple Constraints," Operations Research, 13, 800-814, 1965.

[3] Barlow, R.E. and Proschan, F., Statistical Theory of Reliability and Life Testing, Probability Models, 1975.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: CCS 593	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
A FURTHER GENERALIZED KETTELE ALGORITHM WITH MULTIPLE CONSTRAINTS

5. TYPE OF REPORT & PERIOD COVERED
Technical

6. PERFORMING ORG. REPORT NUMBER
CCS 593

7. AUTHOR(s)
A. Charnes
D. Zhang

8. CONTRACT OR GRANT NUMBER(s)
N00014-86-C-0398

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Center for Cybernetic Studies, UT Austin
Austin, Texas 78712-1177

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
Office of naval Research (Code 434)
Washington, DC

12. REPORT DATE
March 1988

13. NUMBER OF PAGES
21

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
This document has been approved for public release and sale; its distribution is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
reliability, life testing, Kettele Algorithms, optimal redundancy

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)
J.D. Kettele, using dynamic programming, developed a simple algorithm (KA) for the optimal redundancy problem in reliability and life testing problems with a single constraint. F.Proschan and T.B. Bray gave a generalization of Kettele's dynamce programming algorithm to include multiple constraints. To solve a much broader class of optimization problems than in [1], R.E. Barlow and F. Proschan generalized the Kettele algorithm (GKA) to apply to strictly increasing separable function problems with a single constraint.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. Abstract (con't)

In this paper, we consider a still more general optimization model and develop a Further Generalized Kettele Algorithm (FGKA) to apply to multiple constraints, etc. As an example, an integer Lexicographic programming model will be solved (corollary 1, section 2). Furthermore, another form of the more general optimization model is pointed out in section 4 of the paper.